

## **Instructions to use the USER Microservice:**

**To initialize the database use command:**

FLASK\_APP=user flask init

To start all the Microservices use command: foreman start

## **USER Microservice:**

- **To a retrieve User's profile (GET method) use the following curl command:**

```
curl -v 'http://127.0.0.1:5000/api/v1/resources/user?username=user_pavan'
```

**You can authenticate user using 2 ways:**

1. **Using the POST request by passing only username and password in json.**

```
curl -X POST -v http://127.0.0.1:5000/api/v1/resources/user -d '{"username": "user_pavan",  
"password": "12ds"}'
```

**\*NOTE: If other fields are passed in json, then it will be treated as create new user scenario.**

2. **Using the GET request by passing the username and password in the url parameters (not advised because of security concerns):**

```
curl -v 'http://127.0.0.1:5000/api/v1/resources/user?username=user_pavan&password=12ds'
```

- **To create a new User profile use the following curl command (POST method):**

```
curl -X POST -v http://127.0.0.1:5000/api/v1/resources/user -d '{"username": "joker12",  
"display_name": "Joker", "password": "serious", "homepage_url": "joker.com", "email":  
"joker@joker.com"}'
```

- **Change User's Password use the following curl command:**

PATCH: curl -X PATCH -v <http://127.0.0.1:5000/api/v1/resources/user?username=joker12> -d '{ "password": "serious123" }'

- **To delete a User's use the following curl command:**

DELETE: curl -X DELETE -v '[http://127.0.0.1:5000/api/v1/resources/user?username=user\\_pavan](http://127.0.0.1:5000/api/v1/resources/user?username=user_pavan)'

## **TRACKS Microservice:**

- **To create a new Track (POST method) use the following command:**

```
curl -X POST -v http://127.0.0.1:5200/api/v1/resources/tracks -d '{"track_title": "tango", "album_title": "Joker", "artist": "brad", "length": "12:12:12", "track_url": "/home/student/Music/tracks/StrongerThanBefore.mp3", "album_art_url": "tuffy"}'
```

\*NOTE: Add all the field. If no value then leave the double quotes empty. Ex. "".

- **To retrieve all Track's (GET method) use the following curl command:**

```
curl -v 'http://127.0.0.1:5200/api/v1/resources/tracks'
```

- **To retrieve a particular Track (GET method) by passing the track\_url, use the following command:**

```
curl -v 'http://127.0.0.1:5200/api/v1/resources/tracks?track_url=/home/student/Music/tracks/Stronger.mp3'
```

- **To edit a Track (PUT method) use the following command:**

```
curl -X PUT -v http://127.0.0.1:5200/api/v1/resources/tracks?track_url=/home/student/Music/tracks/Stronger.mp3 -d '{"track_title": "JustMe", "album_title": "Joker", "artist": "rf", "length": "12:60:12", "track_url": "/home/student/Music/tracks/Stronger.mp3", "album_art_url": "LetsSee"}'
```

\*NOTE: Add all the field. If no value then leave the double quotes empty. Ex. "".

- **To delete a Track (DELETE method) use the following command:**

```
curl -X DELETE -v 'http://127.0.0.1:5200/api/v1/resources/tracks?track_url=/home/student/Music/tracks/StrongerThanBefore.mp3'
```

## **DESCRIPTION Microservice:**

- **Retrieve a User's Description of a track GET:**

```
curl 'http://127.0.0.1:5100/api/v1/resources/descriptions?username=user_priyanka&track_url=/home/student/Music/tracks/Yeah.mp3'
```

- **To set a User's Description of a track use the following curl command:**

```
curl -X POST -v http://127.0.0.1:5100/api/v1/resources/descriptions -d '{"username":"user_anthony", "track_url":"/home/student/Music/tracks/Yeah.mp3", "description":"Looks good"}'
```

\*NOTE: Add all the field. If no value then leave the double quotes empty. Ex. "".

## **PLAYLIST Microservice:**

- **To create (POST method) a new Playlist use the following command:**

```
curl -X POST -v http://127.0.0.1:5300/api/v1/resources/playlist -d '{ "all_tracks": [  
  {  
    "track_url": "/home/student/Music/tracks/Yeah.mp3"  
  },  
  {  
    "track_url": "/home/student/Music/tracks/I_Gotta_Feeling.mp3"  
  }  
}]'
```

```
], "playlist_title": "NewPlaylistIsThis", "username": "user_priyanka", "description": "This playlist contains some of my songs"}}
```

**\*NOTE:** Track's microservice url can also be directly be placed in the location of track\_url.

Ex. Instead of this: "track\_url": "/home/student/Music/tracks/HULUYeahItsMySong.mp3",  
we can also pass:

"track\_url": "http://127.0.0.1:5200/api/v1/resources/tracks?track\_url=/home/student/Music/tracks/HULUYeahItsMySong.mp3"

ii) Add all the field. If no value then leave the double quotes empty. Ex. "".

- **To retrieve (GET method) a Playlist use the following command:**

```
curl -v 'http://127.0.0.1:5300/api/v1/resources/playlist?  
playlist_title=NewPlaylistIsThis&username=user_priyanka'
```

- **To delete (DELETE method) a Playlist use the following command:**

```
curl -X DELETE -v 'http://127.0.0.1:5300/api/v1/resources/playlist?  
username=user_priyanka&playlist_title=NewPlaylistIsThis'
```

- **To list (GET method) all Playlist's use the following command:**

```
curl -v 'http://127.0.0.1:5300/api/v1/resources/playlist'
```

- **To list (GET method) Playlist's created by a particular User, use the following command:**

```
curl -v 'http://127.0.0.1:5300/api/v1/resources/playlist?username=user_anthony'
```